 Ted Smiley Jr correct fig saving

071146c 9 hours ago

1 contributor

11435 lines (11434 sloc) 856 KB

```

In [1]: import pandas as pd
import csv
import numpy as np
import matplotlib.pyplot as plt
import requests
from pprint import pprint
import scipy.stats as stats

import os
from census import Census
from pprint import pprint
# Census API Key
from config import api_key

buz_info = 44021
zipcode = pd.read_csv("Resources/Public_ZipCode.csv")
zipcode2 = pd.read_csv("Resources/charter_df.csv")
zipcode.head()

```

Out[1]:

	Building Name	District Name	County	City, State, Zip Code	Performance Index Score 2015-16	City	State	Zip Code	Code+4
0	Ada Elementary School	Ada Exempted Village	Hardin	Ada, OH, 45810-1013	96.125	Ada	OH	45810	1013.0
1	Ada High School	Ada Exempted Village	Hardin	Ada, OH, 45810-1013	91.667	Ada	OH	45810	1013.0
2	Sandusky Middle School	Sandusky City	Erie	Sandusky, OH, 44870-2616	62.772	Sandusky	OH	44870	2616.0
3	Meigs Primary School	Meigs Local	Meigs	Middleport, OH, 45760-9717	NC	Middleport	OH	45760	9717.0
4	Meigs Intermediate School	Meigs Local	Meigs	Middleport, OH, 45760-9717	70.394	Middleport	OH	45760	9717.0

```

In [2]: #API CALL
#zipcode[["zipcode", "extra"]] = zipcode["Zip Code"].str.split("-", expand=True)
zcode = zipcode['Zip Code']
zip_pd = pd.DataFrame(zcode)

zcode2 = zipcode2['Zip Code']
zip_pd2 = pd.DataFrame(zcode2)

zip_pd2.head()

```

Out[2]:

	Zip Code
0	45404
1	45207
2	43537
3	44115
4	44663

```

In [3]: zip_pd.replace('(^\s+|\s+$)', '', regex=True, inplace=True)

```

```

In [4]: zcode.head()

```

```

Out[4]: 0    45810
1    45810
2    44870
3    45760
4    45760
Name: Zip Code, dtype: int64

```

```

In [5]: list(zcode)

```

```

Out[5]: [45810,
45810.]

```

45029,  
44870,  
45760,  
45760,  
45013,  
43802,  
43207,  
45656,  
45656,  
45369,  
45503,  
45404,  
45207,  
45631,  
45669,  
43011,  
43901,  
45628,  
43537,  
44122,  
45601,  
44121,  
43604,  
44115,  
44663,  
43344,  
45710,  
44133,  
44857,  
44644,  
43207,  
45414,  
44320,  
44709,  
43604,  
44130,  
44502,  
43609,  
44484,  
45237,  
44883,  
45710,  
44134,  
44111,  
44102,  
44134,  
44134,  
43613,  
44503,  
44817,  
44055,  
45859,  
44256,  
43302,  
43302,  
43430,  
44306,  
44311,  
45850,  
44255,  
44255,  
43138,  
43138,  
43138,  
44030,  
44030,  
44906,  
45601,  
44805,  
44707,  
43920,  
45640,  
43219,  
45850,  
45144,  
45144,  
45651,  
44601,  
45801,  
-----

45212,  
45804,  
44132,  
44102,  
45036,  
45036,  
43229,  
45101,  
43119,  
43227,  
45505,  
43229,  
43207,  
45155,  
44555,  
44134,  
43102,  
44111,  
44123,  
43615,  
45044,  
43229,  
43205,  
43232,  
43204,  
45214,  
44112,  
43102,  
44102,  
45202,  
45410,  
45102,  
45103,  
45224,  
44052,  
43207,  
45103,  
45801,  
43207,  
44485,  
45711,  
45415,  
44509,  
44077,  
43614,  
45044,  
44646,  
44270,  
45236,  
44052,  
44060,  
43231,  
45255,  
44804,  
44011,  
44003,  
43004,  
44120,  
44120,  
44647,  
44120,  
45302,  
45302,  
45303,  
43604,  
45303,  
43212,  
43212,  
45005,  
45237,  
45415,  
43571,  
43615,  
44104,  
43701,  
45813,  
44109,  
44052,  
44052,

43612,  
45813,  
44606,  
44509,  
44103,  
44703,  
44076,  
43228,  
44212,  
43068,  
43068,  
43064,  
45415,  
44804,  
43035,  
44804,  
45304,  
43315,  
45304,  
43502,  
44114,  
43502,  
43232,  
43055,  
44444,  
43607,  
43204,  
45231,  
43609,  
45814,  
45814,  
44321,  
44135,  
43232,  
44805,  
44805,  
43003,  
44004,  
43103,  
45780,  
43420,  
44807,  
44807,  
44201,  
44875,  
44023,  
45873,  
44615,  
44202,  
44202,  
44010,  
44515,  
43026,  
44011,  
44011,  
44011,  
44012,  
43222,  
44708,  
43512,  
43512,  
45067,  
44428,  
45612,  
44473,  
44305,  
44203,  
45784,  
43713,  
43713,  
44054,  
43221,  
44430,  
44145,  
43452,  
45103,  
45103,  
44333,  
45001

45801,  
45801,  
45801,  
45449,  
44140,  
44140,  
44481,  
44122,  
43026,  
43716,  
43716,  
45342,  
43203,  
43224,  
43920,  
43920,  
45434,  
45419,  
44146,  
43213,  
45218,  
44704,  
43906,  
45305,  
45305,  
43311,  
45416,  
43311,  
44811,  
44811,  
44813,  
45420,  
44460,  
45714,  
44118,  
44109,  
44021,  
44401,  
44610,  
44814,  
43155,  
43155,  
45036,  
43209,  
45371,  
45106,  
45371,  
45106,  
44310,  
44305,  
43614,  
45715,  
45431,  
43209,  
45614,  
43074,  
43204,  
43605,  
44087,  
44880,  
45418,  
45107,  
45107,  
43136,  
45682,  
43112,  
45067,  
44450,  
43160,  
44505,  
45236,  
45817,  
45817,  
45895,  
44512,  
44512,  
44223,  
44612,  
44106,

45237,  
44264,  
45306,  
45306,  
44120,  
44118,  
44695,  
43402,  
43402,  
43614,  
44430,  
45308,  
45308,  
44124,  
45244,  
45404,  
44147,  
44147,  
45231,  
44060,  
43912,  
45211,  
43913,  
44240,  
44907,  
44402,  
44402,  
43026,  
43209,  
45371,  
44403,  
44403,  
44144,  
44142,  
44144,  
43235,  
44054,  
44142,  
45309,  
43026,  
44266,  
45320,  
44212,  
44044,  
44124,  
43506,  
43506,  
44122,  
44320,  
44854,  
44077,  
44460,  
44256,  
43207,  
43015,  
45681,  
44820,  
44842,  
43952,  
43723,  
44109,  
44004,  
44514,  
45680,  
43567,  
43204,  
43609,  
44021,  
44822,  
45304,  
45377,  
43723,  
44122,  
45233,  
44212,  
43907,  
45658,  
43920,  
42771

43724,  
43314,  
44112,  
43725,  
43725,  
45311,  
44406,  
45638,  
44241,  
43064,  
44614,  
43110,  
44406,  
44118,  
44707,  
44413,  
44062,  
44062,  
43315,  
43315,  
43316,  
45005,  
45005,  
45005,  
45408,  
43112,  
44615,  
44615,  
45238,  
44146,  
44103,  
44303,  
43209,  
44710,  
43130,  
44077,  
45314,  
45314,  
43207,  
45822,  
45822,  
44444,  
44081,  
43011,  
43011,  
45459,  
43560,  
43140,  
45040,  
44663,  
43545,  
45042,  
44890,  
45692,  
45014,  
44904,  
44240,  
44146,  
44104,  
45801,  
45840,  
44281,  
45215,  
45601,  
44022,  
44022,  
44470,  
44087,  
44112,  
44483,  
44483,  
43203,  
44483,  
44509,  
43230,  
44024,  
44024,  
44109,



44105,  
44721,  
44128,  
45248,  
44617,  
43611,  
45223,  
43081,  
43608,  
43558,  
43055,  
45619,  
44070,  
44610,  
45211,  
45601,  
44141,  
44230,  
43113,  
43113,  
44135,  
44708,  
43207,  
44102,  
43616,  
45662,  
44683,  
44683,  
44813,  
45066,  
43102,  
45066,  
44720,  
44052,  
43055,  
45103,  
45420,  
45013,  
43202,  
44614,  
45113,  
45245,  
44254,  
45239,  
43410,  
43410,  
45692,  
45417,  
45828,  
43214,  
45251,  
45251,  
45251,  
45224,  
44110,  
43316,  
44827,  
43085,  
43050,  
44028,  
44028,  
44883,  
44408,  
44146,  
44004,  
45830,  
45830,  
43224,  
45373,  
43811,  
44287,  
43402,  
44030,  
44695,  
45831,  
45831,  
44319,  
45373,  
45723.

44321,  
44321,  
44028,  
44041,  
45881,  
43812,  
43731,  
45238,  
45318,  
45318,  
45385,  
43616,  
43221,  
45044,  
44827,  
44217,  
44212,  
44408,  
45832,  
44805,  
45832,  
44035,  
44255,  
45806,  
43402,  
43528,  
43420,  
43731,  
45030,  
43614,  
44320,  
44410,  
44403,  
44221,  
44125,  
44125,  
44817,  
44506,  
44104,  
44618,  
43214,  
44053,  
44618,  
43050,  
43222,  
43440,  
43440,  
45107,  
44106,  
44053,  
43014,  
43014,  
43123,  
45238,  
43040,  
44432,  
43015,  
43035,  
43968,  
44833,  
45638,  
45638,  
43062,  
45236,  
44685,  
43017,  
43620,  
44117,  
43220,  
43207,  
45638,  
44105,  
43125,  
45066,  
43512,  
45458,  
45801,  
45417,  
43512

43512,  
44721,  
44052,  
45238,  
44620,  
43515,  
43515,  
43515,  
45377,  
44109,  
44134,  
45177,  
43524,  
43613,  
43229,  
44104,  
43701,  
45345,  
44408,  
45320,  
44514,  
43214,  
45840,  
45319,  
44139,  
43617,  
43031,  
44110,  
45206,  
44622,  
44145,  
44622,  
44875,  
44230,  
45243,  
45429,  
44706,  
44278,  
45408,  
43734,  
43771,  
44624,  
43232,  
44053,  
43219,  
45872,  
45044,  
43351,  
44730,  
44730,  
44044,  
45044,  
45069,  
44110,  
45169,  
43219,  
45177,  
44137,  
44137,  
44137,  
43606,  
43028,  
43021,  
43211,  
43065,  
43065,  
43920,  
43016,  
43026,  
44307,  
44483,  
44484,  
44485,  
44485,  
44107,  
44107,  
44107,  
44107,  
43205,

43845,  
44413,  
44325,  
43607,  
44134,  
44420,  
45640,  
43606,  
45390,  
43614,  
43229,  
43608,  
44906,  
45211,  
43068,  
43055,  
43055,  
43055,  
43055,  
43055,  
45458,  
44052,  
44055,  
45426,  
43460,  
43228,  
44663,  
43050,  
44622,  
45320,  
45469,  
44135,  
43035,  
45885,  
45846,  
43130,  
45331,  
45619,  
45822,  
45690,  
44110,  
43078,  
45701,  
43832,  
43160,  
44662,  
44305,  
45331,  
43085,  
43205,  
44137,  
44104,  
45772,  
45613,  
43232,  
44095,  
45432,  
44683,  
44905,  
44012,  
44210,  
44074,  
43450,  
44067,  
45320,  
44224,  
43028,  
43517,  
43517,  
43611,  
44691,  
45152,  
43040,  
44004,  
44004,  
43204,  
43223,  
43604,  
45416

45662,  
44643,  
44643,  
44094,  
43231,  
43224,  
44839,  
43935,  
44113,  
44805,  
45402,  
43212,  
44846,  
44646,  
44446,  
45807,  
43518,  
43518,  
44212,  
45011,  
45013,  
44903,  
43035,  
43015,  
43035,  
43021,  
43302,  
45219,  
43232,  
45767,  
43031,  
43001,  
44044,  
45807,  
44114,  
44114,  
43605,  
45312,  
44221,  
44707,  
45215,  
43440,  
44256,  
44312,  
45240,  
44401,  
45215,  
43935,  
43567,  
43551,  
43613,  
44125,  
45505,  
44817,  
45216,  
44035,  
44035,  
43081,  
45365,  
45322,  
45322,  
45323,  
44012,  
44133,  
43015,  
44319,  
44035,  
45697,  
43062,  
43213,  
44117,  
44112,  
45410,  
44123,  
44236,  
45631,  
45241,  
43085,

43215,  
43540,  
43045,  
44118,  
44060,  
44408,  
45014,  
43537,  
44837,  
45135,  
45014,  
45135,  
43107,  
43130,  
45669,  
45669,  
45669,  
44313,  
45365,  
45365,  
44662,  
43213,  
44077,  
45220,  
45406,  
43556,  
44114,  
44126,  
43205,  
44256,  
44256,  
43571,  
43147,  
43147,  
43147,  
44312,  
44119,  
45327,  
43616,  
43512,  
43227,  
45118,  
45118,  
44837,  
43116,  
43229,  
45778,  
45011,  
45015,  
45013,  
45013,  
44509,  
45120,  
45224,  
45120,  
45420,  
43973,  
44004,  
45385,  
44308,  
44052,  
45415,  
44120,  
44830,  
44260,  
44260,  
43160,  
43160,  
43215,  
44471,  
45219,  
45013,  
45840,  
44310,  
43223,  
45231,  
44074,  
44074,  
44313.

```
...,
44301,
44256,
45623,
44138,
44820,
44270,
45895,
45895,
44224,
44854,
44138,
44515,
45324,
45239,
45215,
44130,
45867,
44113,
45373,
44310,
43103,
44320,
43229,
44906,
43214,
45240,
45403,
44507,
45342,
44820,
43202,
43205,
43213,
43232,
44105,
43219,
44041,
44134,
44134,
44106,
44113,
44105,
43701,
44830,
43227,
44883,
44304,
44130,
43068,
45628,
43074,
44507,
45405,
45833,
45358,
...]
```

```
In [6]: # Run Census Search to retrieve data on all zip codes (2013 ACS5 Census)
# See: https://github.com/CommerceDataService/census-wrapper for library documentation
# See: https://gist.github.com/afhaque/60558290d6efd892351c4b64e5c01e9b for labels

# set up a parameters dictionary
ZIPCODE = []
PAYANN = []

for code in zcode:

    # base url
    query_url = f"https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:{code}&key={api_key}"

    print(query_url)

    #response = requests.get(base_url).json()
```

```

#Run requests to grab the JSON at the requested URL

response = requests.get(query_url)
print(response.status_code)

if response.status_code == 200:
    jsonResponse = response.json()
    ZIPCODE.append(jsonResponse[1][0])
    PAYANN.append(jsonResponse[1][2])

#pprint(response)

#ZIPCODE
#print(EMPSZES)
#PAYANN

https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:45810&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:45810&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:44870&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:45760&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:45760&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:45013&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:43802&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:43207&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:45656&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:45656&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:45369&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:45503&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:45404&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:45207&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:45631&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:45669&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:43011&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:43901&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:45628&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:43537&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:44122&key=2af5575ace0b8385c92d40ea1d52fe11407acc24

```



[illegible]

[illegible]

[illegible]

4/30/91 11:00 AM 4/30/91 11:00 AM

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]



[illegible]



[illegible]

[illegible]

[illegible]



[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]





[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]



[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

40e1d52fe1140/acc24



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]





[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]





[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



200

```
In [7]: ZIPCODE2 = []
PAYANN2 = []

for code in zcode2:

    # base url
    query_url = f"https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:{code}&key={api_key}"

    print(query_url)

    #response = requests.get(base_url).json()

    #Run requests to grab the JSON at the requested URL
    response2 = requests.get(query_url)
    print(response2.status_code)

    if response2.status_code == 200:
        jsonResponse2 = response2.json()
        ZIPCODE2.append(jsonResponse2[1][0])
        PAYANN2.append(jsonResponse2[1][2])
```

```
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:45404&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:45207&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:43537&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:44115&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:44663&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:43207&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:45414&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:44320&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:44709&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:43604&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:44130&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:44502&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:43609&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:44484&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:45237&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:44883&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:44134&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:44111&key=2af5575ace0b8385c92d
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]





[illegible]

```

https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:43229&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:43204&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:45426&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:45044&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:43607&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:44135&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:43624&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
204
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:44135&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:44102&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:45415&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:43068&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:43605&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:45406&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:45011&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:45231&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:43232&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:45662&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:43302&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:43078&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:43302&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:44052&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:43337&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200
https://api.census.gov/data/2016/zbp?get=ZIPCODE,EMPSZES,PAYANN&for=zipcode:45801&key=2af5575ace0b8385c92d40ea1d52fe11407acc24
200

```

```

In [8]: # create dataframe
public_payroll_by_ZIP_df = pd.DataFrame({"ZIPCODE": ZIPCODE,"Total Annual Payroll": PAYANN})
public_payroll_by_ZIP_df.tail()

```

```

Out[8]:

```

	Total Annual Payroll	ZIPCODE
3363	26602	44287
3364	832214	43123

3365	6764	45813
3366	38920	43762
3367	165770	45640

```
In [9]: charter_payfoll_by_ZIP_df = pd.DataFrame({"ZIPCODE": ZIPCODE2,"Total Annual Payroll": PAYANN2})
charter_payfoll_by_ZIP_df.tail()
```

Out[9]:

	Total Annual Payroll	ZIPCODE
270	299452	43078
271	743805	43302
272	124384	44052
273	0	43337
274	685344	45801

```
In [10]: csvpath = "Resources/Public_School_Data.csv"
raw_data = pd.read_csv(csvpath)
charter_schools_data = pd.read_csv("Resources/1617_CS_ACHIEVEMENT.csv")
```

```
In [11]: #Public and Charter School Datat
public_df = raw_data[["Building Name", "District Name", "County", "City, State, Zip Code", "Performance Index Score 2015-16"]]
charter_schools_df = charter_schools_data.loc[:, ['Building Name', 'District Name', 'County', 'City and Zip Code', 'Performance Index Percent 2016-17']]
```

```
In [12]: charter_schools_df[["City", "State", "Zip Code"]] = charter_schools_df["City and Zip Code"].str.split(",",
expand=True).rename(columns=lambda x: f"string_{x+1}")
charter_schools_df.head()
```

Out[12]:

	Building Name	District Name	County	City and Zip Code	Performance Index Percent 2016-17	City	State	Zip Code
0	Pathway School of Discovery	Pathway School of Discovery	Montgomery	Dayton, OH, 45404-2123	66.2	Dayton	OH	45404-2123
1	Alliance Academy of Cincinnati	Alliance Academy of Cincinnati	Hamilton	Cincinnati, OH, 45207-1644	53.8	Cincinnati	OH	45207-1644
2	Wildwood Environmental Academy	Wildwood Environmental Academy	Lucas	Maumee, OH, 43537-1374	61.7	Maumee	OH	43537-1374
3	Ohio Connections Academy, Inc	Ohio Connections Academy, Inc	Cuyahoga	Cleveland, OH, 44115-2229	64.7	Cleveland	OH	44115-2229
4	Quaker Digital Academy	New Philadelphia City	Tuscarawas	New Philadelphia, OH, 44663-2150	54.5	New Philadelphia	OH	44663-2150

```
In [13]: charter_schools_df[["City", "State", "Zip Code"]] = charter_schools_df["City and Zip Code"].str.split(",",
expand=True).rename(columns=lambda x: f"string_{x+1}")
charter_schools_df[["Zip Code", "Code+4"]] = charter_schools_df["Zip Code"].str.split("-",expand=True).rename(columns=lambda x: f"string_{x+1}")
charter_schools_df.head()
```

Out[13]:

	Building Name	District Name	County	City and Zip Code	Performance Index Percent 2016-17	City	State	Zip Code	Code+4
0	Pathway School of Discovery	Pathway School of Discovery	Montgomery	Dayton, OH, 45404-2123	66.2	Dayton	OH	45404	2123
1	Alliance Academy of Cincinnati	Alliance Academy of Cincinnati	Hamilton	Cincinnati, OH, 45207-1644	53.8	Cincinnati	OH	45207	1644
2	Wildwood Environmental Academy	Wildwood Environmental Academy	Lucas	Maumee, OH, 43537-1374	61.7	Maumee	OH	43537	1374

3	Ohio Connections Academy, Inc	Ohio Connections Academy, Inc	Cuyahoga	Cleveland, OH, 44115-2229	64.7	Cleveland	OH	44115	2229
4	Quaker Digital Academy	New Philadelphia City	Tuscarawas	New Philadelphia, OH, 44663-2150	54.5	New Philadelphia	OH	44663	2150

```
In [14]: public_df[["City","State", "Zip Code"]] = public_df["City, State, Zip Code"].str.split(",",expand=True)
public_df[["Zip Code", "Code+4"]] = public_df["Zip Code"].str.split("-",expand=True)
public_df.head()
```

/Users/theodoresmiley/anaconda/lib/python3.6/site-packages/pandas/core/frame.py:2450: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>

```
self[k1] = value[k2]
```

```
Out[14]:
```

	Building Name	District Name	County	City, State, Zip Code	Performance Index Score 2015-16	City	State	Zip Code	Code+4
0	Ada Elementary School	Ada Exempted Village	Hardin	Ada, OH, 45810-1013	96.125	Ada	OH	45810	1013
1	Ada High School	Ada Exempted Village	Hardin	Ada, OH, 45810-1013	91.667	Ada	OH	45810	1013
2	Sandusky Middle School	Sandusky City	Erie	Sandusky, OH, 44870-2616	62.772	Sandusky	OH	44870	2616
3	Meigs Primary School	Meigs Local	Meigs	Middleport, OH, 45760-9717	NC	Middleport	OH	45760	9717
4	Meigs Intermediate School	Meigs Local	Meigs	Middleport, OH, 45760-9717	70.394	Middleport	OH	45760	9717

```
In [15]: public_df.to_csv("Resources/Public_ZipCode.csv", index=False, header=True)
```

```
In [16]: public_payroll= public_payroll_by_ZIP_df.rename(columns={"ZIPCODE": "Zip Code"})
public_payroll.head()
```

```
Out[16]:
```

	Total Annual Payroll	Zip Code
0	77875	45810
1	77875	45810
2	894118	44870
3	9171	45760
4	9171	45760

```
In [17]: public_df.head()
```

```
Out[17]:
```

	Building Name	District Name	County	City, State, Zip Code	Performance Index Score 2015-16	City	State	Zip Code	Code+4
0	Ada Elementary School	Ada Exempted Village	Hardin	Ada, OH, 45810-1013	96.125	Ada	OH	45810	1013
1	Ada High School	Ada Exempted Village	Hardin	Ada, OH, 45810-1013	91.667	Ada	OH	45810	1013
2	Sandusky Middle School	Sandusky City	Erie	Sandusky, OH, 44870-2616	62.772	Sandusky	OH	44870	2616
3	Meigs Primary School	Meigs Local	Meigs	Middleport, OH, 45760-9717	NC	Middleport	OH	45760	9717
	Meigs			Middlenort OH					

4	Intermediate School	Meigs Local	Meigs	Middleport, OH, 45760-9717	70.394	Middleport	OH	45760	9717
---	---------------------	-------------	-------	----------------------------	--------	------------	----	-------	------

```
In [18]: public_df["Zip Code"] = public_df["Zip Code"].astype(int)
public_payroll["Zip Code"] = public_payroll["Zip Code"].astype(int)
public_payroll.dtypes
```

/Users/theodoresmiley/anaconda/lib/python3.6/site-packages/ipykernel\_launcher.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>

"""Entry point for launching an IPython kernel.

```
Out[18]: Total Annual Payroll    object
Zip Code                        int64
dtype: object
```

```
In [19]: #Problems Merging
public_pay_df = public_df.merge(public_payroll, on="Zip Code")
public_pay_df.head()
```

```
Out[19]:
```

	Building Name	District Name	County	City, State, Zip Code	Performance Index Score 2015-16	City	State	Zip Code	Code+4	Total Annual Payroll
0	Ada Elementary School	Ada Exempted Village	Hardin	Ada, OH, 45810-1013	96.125	Ada	OH	45810	1013	77875
1	Ada Elementary School	Ada Exempted Village	Hardin	Ada, OH, 45810-1013	96.125	Ada	OH	45810	1013	77875
2	Ada High School	Ada Exempted Village	Hardin	Ada, OH, 45810-1013	91.667	Ada	OH	45810	1013	77875
3	Ada High School	Ada Exempted Village	Hardin	Ada, OH, 45810-1013	91.667	Ada	OH	45810	1013	77875
4	Sandusky Middle School	Sandusky City	Erie	Sandusky, OH, 44870-2616	62.772	Sandusky	OH	44870	2616	894118

```
In [20]: charter_payroll = charter_payroll_by_ZIP_df.rename(columns={"ZIPCODE": "Zip Code"})
charter_schools_df["Zip Code"] = charter_schools_df["Zip Code"].astype(int)
charter_payroll["Zip Code"] = charter_payroll["Zip Code"].astype(int)

charter_pay_df = pd.merge(charter_schools_df, public_payroll, on="Zip Code")
charter_pay_df.head()
```

```
Out[20]:
```

	Building Name	District Name	County	City and Zip Code	Performance Index Percent 2016-17	City	State	Zip Code	Code+4	Total Annual Payroll
0	Pathway School of Discovery	Pathway School of Discovery	Montgomery	Dayton, OH, 45404-2123	66.2	Dayton	OH	45404	2123	615078
1	Pathway School of Discovery	Pathway School of Discovery	Montgomery	Dayton, OH, 45404-2123	66.2	Dayton	OH	45404	2123	615078
2	Pathway School of Discovery	Pathway School of Discovery	Montgomery	Dayton, OH, 45404-2123	66.2	Dayton	OH	45404	2123	615078
3	Alliance Academy of Cincinnati	Alliance Academy of Cincinnati	Hamilton	Cincinnati, OH, 45207-1644	53.8	Cincinnati	OH	45207	1644	137872
4	Alliance Academy of Cincinnati	Alliance Academy of Cincinnati	Hamilton	Cincinnati, OH, 45207-1644	53.8	Cincinnati	OH	45207	1644	137872

4	Academy of Cincinnati	Academy of Cincinnati	Hamilton	Cincinnati, OH, 45207-1644	53.8	Cincinnati	OH	45207	1644	137872
---	-----------------------	-----------------------	----------	----------------------------	------	------------	----	-------	------	--------

```
In [21]: public_df = public_df.loc[public_df["Performance Index Score 2015-16"] != "NC"]
charter_schools_df = charter_schools_df.loc[charter_schools_df["Performance Index Percent 2016-17"] != "NC"]
public_df["Performance Index Score 2015-16"] = public_df["Performance Index Score 2015-16"].astype(float)
charter_schools_df["Performance Index Percent 2016-17"] = charter_schools_df["Performance Index Percent 2016-17"].astype(float)
```

```
In [22]: #Ethnicity/Race Data
csvpath2 = "Resources/Ethnicity.csv"
raw_data_race = pd.read_csv(csvpath2)
ethnic_df = pd.read_csv(csvpath2)

/Users/theodoresmiley/anaconda/lib/python3.6/site-packages/IPython/core/interactiveshell.py:2717: DtypeWarning: Columns (40) have mixed types. Specify dtype option on import or set low_memory=False.
interactivity=interactivity, compiler=compiler, result=result)
```

In [ ]:

```
In [23]: ### Replace the >95
race_df = raw_data_race[["Building Name", "Student Group", "% of Total Enrollment"]]
race_df = race_df.replace(to_replace = "NC", value = 0)
race_df["% of Total Enrollment"] = race_df["% of Total Enrollment"].astype(str)
race_df["% of Total Enrollment"] = race_df["% of Total Enrollment"].replace(to_replace = r'>95', value = 95.1, regex=True)
race_df["% of Total Enrollment"] = race_df["% of Total Enrollment"].astype(float)
```

```
In [24]: df_white = race_df.loc[race_df["Student Group"] == "White", :]
df_latinx = race_df.loc[race_df["Student Group"] == "Hispanic", :]
df_black = race_df.loc[race_df["Student Group"] == "Black", :]
```

```
In [25]: race_df.head(30)
```

Out[25]:

	Building Name	Student Group	% of Total Enrollment
0	Ada Elementary School	American Indian or Alaskan Native	0.0
1	Ada Elementary School	Asian or Pacific Islander	0.0
2	Ada Elementary School	Black	3.4
3	Ada Elementary School	Hispanic	0.0
4	Ada Elementary School	Multiracial	0.0
5	Ada Elementary School	White	92.5
6	Ada High School	American Indian or Alaskan Native	0.0
7	Ada High School	Asian or Pacific Islander	0.0
8	Ada High School	Black	0.0
9	Ada High School	Hispanic	0.0
10	Ada High School	Multiracial	0.0
11	Ada High School	White	94.1
12	Sandusky Middle School	American Indian or Alaskan Native	0.0
13	Sandusky Middle School	Asian or Pacific Islander	0.0
14	Sandusky Middle School	Black	38.1
15	Sandusky Middle School	Hispanic	4.7
16	Sandusky Middle School	Multiracial	18.8
17	Sandusky Middle School	White	37.9
18	Meigs Primary School	American Indian or Alaskan Native	0.0
19	Meigs Primary School	Asian or Pacific Islander	0.0
20	Meigs Primary School	Black	0.0
21	Meigs Primary School	Hispanic	0.0

22	Meigs Primary School	Multiracial	3.7
23	Meigs Primary School	White	94.5
24	Meigs Intermediate School	American Indian or Alaskan Native	0.0
25	Meigs Intermediate School	Asian or Pacific Islander	0.0
26	Meigs Intermediate School	Black	0.0
27	Meigs Intermediate School	Hispanic	0.0
28	Meigs Intermediate School	Multiracial	0.0
29	Meigs Intermediate School	White	95.1

```
In [26]: public_race_white = df_white.merge(public_df, on="Building Name")
public_race_latinx = df_latinx.merge(public_df, on="Building Name")
public_race_black = df_black.merge(public_df, on="Building Name")

charterwhite_df = pd.merge(charter_schools_df, df_white, on="Building Name")
charterblack_df = pd.merge(charter_schools_df, df_black, on="Building Name")
charterhispanic_df = pd.merge(charter_schools_df, df_latinx, on="Building Name")
```

```
In [27]: #Econ Data
csvpath3 = "Resources/Economic.csv"
raw_data_econ = pd.read_csv(csvpath3)
raw_data_econ.head()
```

Out[27]:

	Building IRN	Building Name	District IRN	District Name	County	Region	Open/Closed Status as of 9/12/2017	School Type	Student Group	Read 3rd Grade 2016- 2017 % Proficient or above	...	Ge En Cc 20 20 Pr or
0	59	Ada Elementary School	45187	Ada Exempted Village	Hardin	Region 6	Open	Elementary School	Disadvantaged	77.4	...	NC
1	59	Ada Elementary School	45187	Ada Exempted Village	Hardin	Region 6	Open	Elementary School	NonDisadvantaged	92.3	...	NC
2	67	Ada High School	45187	Ada Exempted Village	Hardin	Region 6	Open	High School	Disadvantaged	NC	...	56
3	67	Ada High School	45187	Ada Exempted Village	Hardin	Region 6	Open	High School	NonDisadvantaged	NC	...	62
4	83	Sandusky Middle School	44743	Sandusky City	Erie	Region 2	Open	Middle School	Disadvantaged	NC	...	NC

5 rows × 41 columns

```
In [28]: econ_df = raw_data_econ[["Building Name", "Student Group", "% of Total Enrollment"]]
econ_df = econ_df.replace(to_replace = "NC", value = 0)
econ_df["% of Total Enrollment"] = econ_df["% of Total Enrollment"].astype(str)
econ_df["% of Total Enrollment"] = econ_df["% of Total Enrollment"].replace(to_replace = r'>95', value = 95.1, regex=True)
econ_df["% of Total Enrollment"] = econ_df["% of Total Enrollment"].astype(float)
```

```
In [29]: df_poor = econ_df.loc[econ_df["Student Group"] == "Disadvantaged", :]
df_notpoor = econ_df.loc[econ_df["Student Group"] == "NonDisadvantaged", :]
```

```
In [30]: public_poor_df = df_poor.merge(public_df, on="Building Name")
public_notpoor_df = df_notpoor.merge(public_df, on="Building Name")

charter_disadvantaged = pd.merge(charter_schools_df, df_poor, on="Building Name")
```



```
charter_disadvantaged = pd.merge(charter_schools_df, df_poor, on= 'Building Name')
charter_nondisadvantaged = pd.merge(charter_schools_df, df_notpoor, on="Building Name")
```

```
In [31]: #Data to analyze
public_black_cuya = public_race_black.loc[public_race_black["County"] == "Cuyahoga", :]
public_white_cuya = public_race_white.loc[public_race_white["County"] == "Cuyahoga", :]
public_latinx_cuya = public_race_latinx.loc[public_race_latinx["County"] == "Cuyahoga", :]
public_poor_cuya = public_poor_df.loc[public_poor_df["County"] == "Cuyahoga", :]
public_notpoor_cuya = public_notpoor_df.loc[public_notpoor_df["County"] == "Cuyahoga", :]

charter_nondisadv_cuyahoga = charter_nondisadvantaged.loc[charter_nondisadvantaged["County"] == "Cuyahoga"]
charter_disadv_cuyahoga = charter_disadvantaged.loc[charter_disadvantaged["County"] == "Cuyahoga"]
charterwhite_cuyahoga = charterwhite_df.loc[charterwhite_df["County"] == "Cuyahoga"]
charterblack_cuyahoga = charterblack_df.loc[charterblack_df["County"] == "Cuyahoga"]
charterhispanic_cuyahoga = charterhispanic_df.loc[charterhispanic_df["County"] == "Cuyahoga"]

public_pay_cuya = public_pay_df.loc[public_pay_df["County"] == "Cuyahoga"]
charter_pay_cuya = charter_pay_df.loc[charter_pay_df["County"] == "Cuyahoga"]
```

```
In [32]: public_pay_cuya["Performance Index Score 2015-16"] = public_pay_cuya["Performance Index Score 2015-16"].replace(to_replace = "NC", value = 0)

/Users/theodoresmiley/anaconda/lib/python3.6/site-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
    """Entry point for launching an IPython kernel.
```

```
In [33]: charter_pay_cuya["Performance Index Percent 2016-17"] = charter_pay_cuya["Performance Index Percent 2016-17"].replace(to_replace = "NC", value = 0)

/Users/theodoresmiley/anaconda/lib/python3.6/site-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
    """Entry point for launching an IPython kernel.
```

```
In [34]: public_pay_cuya["Performance Index Score 2015-16"] = public_pay_cuya["Performance Index Score 2015-16"].astype(float)
#public_pay_cuya["Performance Index Score 2015-16"]

/Users/theodoresmiley/anaconda/lib/python3.6/site-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
    """Entry point for launching an IPython kernel.
```

```
In [35]: charter_pay_cuya["Performance Index Percent 2016-17"] = charter_pay_cuya["Performance Index Percent 2016-17"].astype(float)
#public_pay_cuya["Performance Index Score 2015-16"]

/Users/theodoresmiley/anaconda/lib/python3.6/site-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
    """Entry point for launching an IPython kernel.
```

```
In [ ]:
```

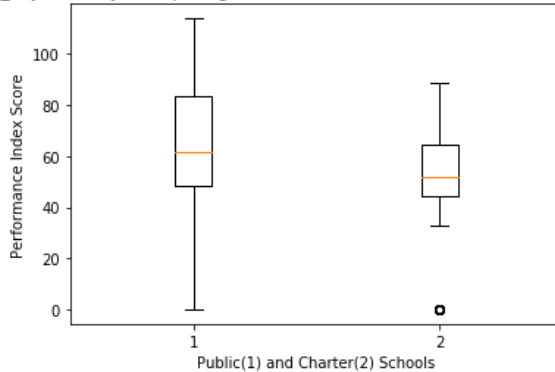
```
In [67]: # t-test
public_df_cuya = public_df.loc[public_df["County"] == "Cuyahoga"]
charter_df_cuya = charter_schools_df.loc[charter_schools_df["County"] == "Cuyahoga"]
s1=public_pay_cuya["Performance Index Score 2015-16"]
s2=charter_pay_cuya["Performance Index Percent 2016-17"]
```

```
plt.boxplot([s1, s2])
plt.title("charter_df_cuya County: Comparing Performance between Public(1) and Charter(2) Schools")
plt.xlabel("Public(1) and Charter(2) Schools")
plt.ylabel("Performance Index Score")

fig1 = plt.gcf()
plt.show()
```

/Users/theodoresmiley/anaconda/lib/python3.6/site-packages/numpy/core/fromnumeric.py:57: FutureWarning: reshape is deprecated and will raise in a subsequent release. Please use .values.reshape(...) instead  
 return getattr(obj, method)(\*args, \*\*kwargs)

charter\_df\_cuya County: Comparing Performance between Public(1) and Charter(2) Schools



```
In [68]: fig1.tight_layout()
fig1.savefig("Cuyahoga_Public_Charter_Box.png")
#plt.show()
```

```
In [37]: (t_stat, p) = stats.ttest_ind(s1, s2, equal_var=False)
print(t_stat)
print(p)
```

14.6783582312  
 9.70256904825e-46

```
In [69]: # Public Black Regression
```

```
x_cuy_blk = public_black_cuya["% of Total Enrollment"]
y_cuy_blk = public_black_cuya["Performance Index Score 2015-16"]
```

```
(cbk_slope, cbk_int, cbk_c_r, cbk_p, cbk_std_err) = stats.linregress(x_cuy_blk, y_cuy_blk)
fit = cbk_slope * x_cuy_blk + cbk_int
```

```
fig, ax = plt.subplots()
```

```
fig.suptitle("Relationship Between Ethnicity and Public School Performance Score (Black Students)", fontsize=16, fontweight="bold")
```

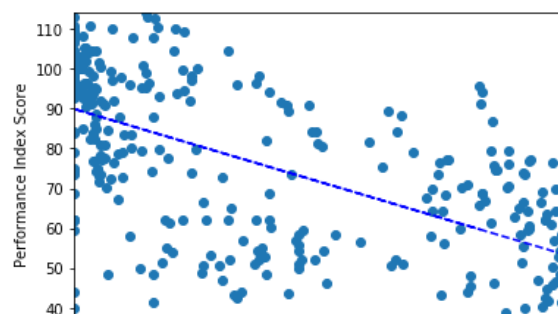
```
ax.set_xlim(min(x_cuy_blk), max(x_cuy_blk))
ax.set_ylim(min(y_cuy_blk), max(y_cuy_blk))
```

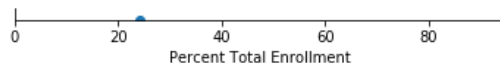
```
ax.set_xlabel("Percent Total Enrollment")
ax.set_ylabel("Performance Index Score")
```

```
ax.plot(x_cuy_blk, y_cuy_blk, linewidth=0, marker='o')
ax.plot(x_cuy_blk, fit, 'b--')
```

```
fig2 = plt.gcf()
plt.show()
```

### Relationship Between Ethnicity and Public School Performance Score (Black Students)





```
In [70]: fig2.tight_layout()
fig2.savefig("Cuyahoga_Public_Black.png")
plt.show()
```

```
In [71]: #Charter School Black Performance

x_cuy_blk2 =charterblack_cuyahoga["% of Total Enrollment"]
y_cuy_blk2 =charterblack_cuyahoga["Performance Index Percent 2016-17"]

(cbk2_slope, cbk2_int, cbk2_c_r, cbk2_p, cbk2_std_err) = stats.linregress(x_cuy_blk2, y_cuy_blk2)
fit = cbk2_slope * x_cuy_blk2 + cbk2_int

fig, ax = plt.subplots()

fig.suptitle("Relationship Between Ethnicity and Charter School Performance Score (Black Students)", fontsize=16, fontweight="bold")

ax.set_xlim(min(x_cuy_blk2), max(x_cuy_blk2))
ax.set_ylim(min(y_cuy_blk2), max(y_cuy_blk2))

ax.set_xlabel("Percent Total Enrollment")
ax.set_ylabel("Performance Index Score")

ax.plot(x_cuy_blk2, y_cuy_blk2, linewidth=0, marker='o')
ax.plot(x_cuy_blk2, fit, 'b--')

fig3 = plt.gcf()
```